

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

## Zadání bakalářské práce

Student:

**Martin Urbánek**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe  
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: KVADOS, a.s.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Ing. Michal Krumnikl, Ph.D.**

Konzultant bakalářské práce: Bc. Radek Pindora

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 26. dubna 2017

.....  
Molárnek

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 26. dubna 2017

  
.....

Rád bych poděkoval panu Mgr. Ing. Michalu Krumníkovi, Ph.D. za pomoc při vypracování této práce. Dále bych chtěl poděkovat lidem z firmy Kvados a.s. za umožnění vykonání bakalářské praxe a veškerou pomoc při jejím absolvování. Děkuji také rodičům a přítelkyni za podporu při studiu.

## **Abstrakt**

V této bakalářské práci popisuji průběh individuální bakalářské praxe ve společnosti Kvados a.s. Práce obsahuje informace o odborném zaměření firmy, zadáných úkolech, jejich implementaci a technologie využité při řešení. V poslední části jsou popsány získané vědomosti a schopnosti a celkové zhodnocení přínosů absolvování praxe.

**Klíčová slova:** Kvados a.s., Android, synchronizace dat, odborná praxe

## **Abstract**

In this bachelor thesis I describe process of individual professional practice in the company Kvados a.s. The thesis contains information about specialization of the company, my assigned tasks, their implementation and technologies used in solving. The last part describes acquired knowledge and skills and overall assessment of benefits gained by professional practice.

**Key Words:** Kvados a.s., Android, data synchronization, professional practice

# Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam tabulek	10
<b>1 Úvod</b>	<b>12</b>
<b>2 Popis společnosti a mé pracovní zařazení</b>	<b>13</b>
2.1 O firmě . . . . .	13
2.2 Pracovní zařazení, náplň a prostředí . . . . .	13
<b>3 Použité technologie</b>	<b>14</b>
3.1 Visual Studio . . . . .	14
3.2 XML . . . . .	14
3.3 SQL . . . . .	14
3.4 Současný vývoj mobilních aplikací ve firmě . . . . .	15
<b>4 Seznam zadaných úkolů a časová náročnost</b>	<b>16</b>
<b>5 Hlavní úkol</b>	<b>17</b>
5.1 Vstup . . . . .	17
5.2 Výstup . . . . .	19
5.3 Analýza a návrh . . . . .	19
5.4 Implementace . . . . .	22
<b>6 Další úkoly</b>	<b>26</b>
6.1 První úkol . . . . .	26
6.2 Kopírování zdrojů . . . . .	26
6.3 Rozvoj bezpečnosti . . . . .	26
6.4 Budoucí vývoj aplikací . . . . .	27
<b>7 Využité dovednosti získané v průběhu studia</b>	<b>28</b>
<b>8 Chybějící znalosti</b>	<b>29</b>
<b>9 Závěr</b>	<b>30</b>
<b>Literatura</b>	<b>31</b>

## Seznam použitých zkratk a symbolů

CRM	– Customer Relationship Management – řízení vztahů se zákazníky
ERP	– Enterprise Resource Planning – plánování podnikových zdrojů
MSBuild	– Microsoft Build Tools
TFS	– Team Foundation Server
XML	– Extensible Markup Language
SQL	– Structured query language
IDE	– Integrated Development Environment
GUI	– Graphical User Interface



## Seznam obrázků

1	Role Trandef souboru . . . . .	16
2	Manipulace s Trandefem . . . . .	20
3	Struktura projektů . . . . .	21
4	Třidní diagram . . . . .	23
5	GUI testovací aplikace . . . . .	25

## Seznam tabulek

1	Vyjádření časové náročnosti . . . . .	16
---	---------------------------------------	----

## Seznam výpisů zdrojového kódu

1	Element Extension . . . . .	17
2	Element DataWorkerUsing . . . . .	18
3	Element Parameter . . . . .	19
4	Integrace MSBuildu . . . . .	24

# 1 Úvod

VŠB-TU Ostrava umožňuje studentům nabýt praktické znalosti a zkušenosti formou bakalářské praxe. Rozhodl jsem se toho využít a spojit tak práci se školou. Tématem této bakalářské práce je tedy popis průběhu odborné praxe ve společnosti Kvados a.s.

Do firmy jsem nastoupil na pozici Android programátor, kde jsem se připojil k týmu vývojářů. Hlavním cílem bylo udělat skupinu aplikací, které usnadní práci se souborem, který definuje synchronizační proces v mobilní aplikaci MyAvis NG. Kromě toho jsem se také podílel na dalších úkolech. Tyto úkoly byly spojeny s vývojem tohoto produktu.

Následující kapitoly obsahují informace o firmě, pracovní náplni, použitých technologiích, popis řešení jednotlivých úkolů a celkové zhodnocení praxe.

## 2 Popis společnosti a mé pracovní zařazení

V následujících podkapitolách popíši charakteristiku firmy, produkty se kterými jsem přišel do styku a pracovní zařazení ve společnosti.

### 2.1 O firmě

KVADOS je od roku 1992 respektovaným výrobcem a dodavatelem vlastních softwarových řešení, služeb a poradenství [1]. Základní rysy, na kterých společnost staví, jsou kvalitně řízené projekty a schopnost porozumět procesům a působením klientů. Firma uplatňuje své produkty ve 12 zemích světa a zaměstnává 150 pracovníků. V současné době má firma v nabídce 9 produktů, které klienty podporují v jejich podnikání a v řízení a podpoře efektivity procesů.

- VENTUS®- ERP systém pro komplexní řešení procesů středních a velkých společností, se specializací na velkoobchod, maloobchod a logistiku ve skladech, orientovaný na zákaznická přizpůsobení a dodaný přímo výrobcem.
- myAvisNG®- Komplexní řešení CRM pro správu a řízení obchodních, marketingových a distribučních činností, s podporou práce v terénu, pro efektivní a pohodlnou práci na tabletech s operačním systémem Android. Přináší nejen rozsáhlé možnosti ke komplexní prezentaci zboží a služeb, ale také k pohodlnému a snadnému pořizování a zpracování dat.

Mezi další produkty se řadí myTEAM®, myFABER®, myCASH®, myWORK®, mySTOCK®, myDATACENTER®, myACADEMY®.

### 2.2 Pracovní zařazení, náplň a prostředí

Do firmy jsem nastoupil na pozici Android programátor, kde jsem se přidal do týmu 6 dalších spolupracovníků. Mojí hlavní pracovní náplní bylo naprogramovat skupinu aplikací, která usnadní manipulaci se synchronizačním souborem. Také jsem se podílel na vývoji produktu MyAvis NG, který byl hlavní pracovní náplní našeho týmu.

Při nástupu mi byl přidělen osobní počítač a tablet, na kterém jsem ladil a testoval své změny a inovace v aplikaci MyAvis NG. Každé ráno ve stejnou hodinu jsme měli Stand-up meetingy, což se používá v metodě agilního vývoje softwaru - Scrumu. Na těchto schůzkách jsme zejména řešili zadávání úkolů pro jednotlivé vývojáře. Využíval jsem také produkt VENTUS, kde jsem spravoval svou personalistiku. Komunikaci tvořil zejména osobní kontakt, email či Skype. Samotná práce probíhala v kanceláři, kde jsem pracoval s mým mentorem a dalšími 2 spolupracovníky.

## 3 Použité technologie

Tato sekce se věnuje technologiím, které byly nutnou součástí při řešení úkolů. Nejprve jsou uvedeny technologie spjaté s vývojem optimalizace synchronizačního souboru. Poslední podkapitola obsahuje informace o vývoji aplikace MyAvis NG.

### 3.1 Visual Studio

Visual Studio je sada vývojářských nástrojů od společnosti Microsoft. Je používáno k vývoji programů na operačním systému Windows, ale také s ním můžeme vyvíjet mobilní a webové aplikace.

- **C#** je programovací jazyk, který je navržen pro vytváření různorodých aplikací, které běží na frameworku .NET. C# je jednoduchý, výkonný, typově bezpečný a objektově orientovaný. Řada inovací v jazyce C# umožňuje rychlý vývoj aplikací při zachování expresivity a elegance jazyků stylu C.[3]
- **MSBuild** (nebo také Microsoft Build Tools či Microsoft Build Engine) je nástroj, který pomáhá automatizovat proces vytváření softwaru, zahrnuje kompilování zdrojového kódu, packaging, testování, a vytváření dokumentací. Jedná se o skriptovací jazyk využívající syntaxi XML. S MSBuildem je možné sestavovat Visual Studio projekty bez instalace samotného VS. Navíc je k dispozici zdarma.
- **TFS** Team Foundation Server je produkt společnosti Microsoft, který poskytuje nástroje využitelné k managementu zdrojového kódu, automatickému sestavování, vykazování, správě projektů, testování a release management. TFS pokrývá značnou část vývojového cyklu produktu.

### 3.2 XML

Extensible Markup Language (XML) je jednoduchý, velmi flexibilní textový formát, který hraje velmi důležitou roli při výměně nejrůznějších dat na Webu a jinde. Byl vyvinut a standardizován konsorciem W3C.

### 3.3 SQL

Structured query language (SQL) je standardizovaný dotazovací jazyk pro ukládání, manipulaci a získávání dat z databází.[2]

### 3.4 Současný vývoj mobilních aplikací ve firmě

V této kapitole je popsán vývoj aplikace MyAvisNG. Technologie a principy uplatňované při vývoji aplikace jsou napsány zjednodušeně a stručně z důvodu rozsáhlosti problematiky a tyto metody jsem primárně nevyužíval při práci na hlavním úkolu této bakalářské praxe.

Mobilní aplikace na platformě Android se ve firmě programuje jak v Android Studiu tak také ve Visual Studiu. Vychází se z metod, principů a tříd, které jsou používány pro psaní WinForms SmartClient aplikací. Využívá se princip tzv. metadat, která jsou uložena ve formátu XML a kompilovatelná Visual Studiemi.

Vzhled a chování jedné aktivity definuje uzel WorkItem, který je definovaný v metadatech. Celá aplikace se tak skládá z množiny WorkItemů. Ve WorkItemu můžeme definovat např. tyto funkcionality:

- **Views** - popisuje UI WorkItemu.
- **Commands** - Command představuje akci, která se na WorkItemu může spustit.
- **Extension** - rozšiřují funkčnost WorkItemu a jsou vyvolány pomocí triggerů.

#### 3.4.1 Android Studio

Android studio je oficiální IDE pro vyvíjení Android aplikací, založené na IntelliJ IDEA. Android Studio je úzce spajto s nástrojem pro automatizaci sestavování programu Gradle. Také obsahuje emulátor s množstvím funkcí.

#### 3.4.2 Jádro aplikace

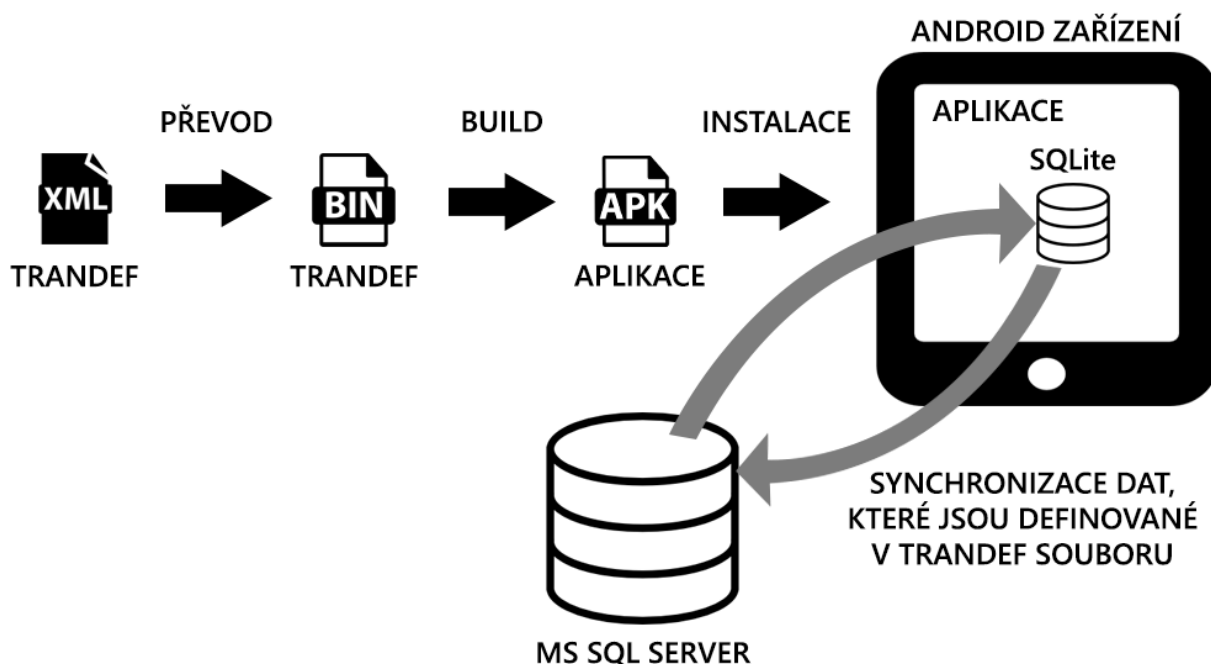
Jádro aplikace je základní projekt, který je samostatně spustitelný na zařízení s operačním systémem Android a obsahuje nejzákladnější moduly, které jsou součástí zákaznických projektů. Zákaznické projekty tedy tvoří nadstavbu jádra aplikace.

#### 3.4.3 Zákaznické projekty

Každý zákaznický projekt ve firmě obsahuje různá prostředí. Každé z těchto prostředí může mít své vlastní nastavení, zdrojový kód a využívat jiné zdroje. Zejména se jedná o vývojářské, testovací a produkční prostředí, ale může obsahovat i jiné. Android Studio umožňuje vytvářet tyto varianty sestavení projektu.

## 4 Seznam zadaných úkolů a časová náročnost

Cílem mé odborné praxe bylo udělat množinu aplikací, které optimalizují práci s Trandef souborem, který nám definuje, jaká data se budou synchronizovat mezi Android klientem a Microsoft SQL Serverem. Na obrázku 1 můžeme vidět proces synchronizace dat a roli Trandef souboru. Trandef soubor popisují detailněji v následující kapitole.



Obrázek 1: Role Trandef souboru

Tento úkol však nebyl jediným úkolem, na kterém jsem ve společnosti pracoval. Podílel jsem se na vylepšení bezpečnosti Android aplikace. Dále jsem pracoval na menších úkolech, o kterých se zmiňuji v kapitole Vedlejší úkoly.

Dále uvádím přibližnou časovou náročnost na jednotlivých úkolech.

Tabulka 1: Vyjádření časové náročnosti

Činnost	Počet dní
Příprava a zaučení technologií potřebné k vývoji Android aplikace	10
Hlavní úkol - optimalizace práce s Trandef souborem	25
Vedlejší úkoly - bezpečnost Android aplikace a další drobné úkoly	15



## 5 Hlavní úkol

V následujících podkapitolách uvádím informace o vstupech, výstupech, analýze, návrhu a problematice implementace jednotlivých prvků aplikací.

### 5.1 Vstup

Vstup tvoří 2 soubory. Trandef - synchronizační soubor a Parameters - doplňující soubor s parametry. Tyto soubory jsou popsány v následujících podkapitolách.

#### 5.1.1 Trandef soubor

Firma používá označení Trandef pro označení synchronizačního souboru. Jedná se o zkratku z anglického transfer definition. Trandef je XML soubor, který popisuje synchronizační proces neboli stažení a odesílání dat na server/z serveru. Každý projekt může mít vlastní definici synchronizace. Při instalaci aplikace do zařízení se soubor převádí do binárního formátu a následně je přidán do apk souboru. Samotná synchronizace se spouští pomocí grafického rozhraní aplikace, kde uživatel funkci spustí při stisknutí tlačítka "synchronizovat".

Root tvoří element DataIntegrationDefinition, který obsahuje 5 základních elementů (Extensions, Preprocessors, Publisher, Subscriber a Tables).

V elementech Extensions, Publisher a Subscriber jsou definice tříd a jejich vstupních parametrů, které se při synchronizaci inicializují a provedou. Elementy Preprocessors a Tables obsahují informace o samotných entitách a nesou více než 99% obsahu souboru. Tyto Elementy upravují samotné data obsažené v databázích.

---

```
<Extension>
  <Arguments />
  <TypeName>SyncCallsCalendarEventsDIExtension</TypeName>
  <Using>Subscriber</Using>
</Extension>
```

---

Výpis 1: Element Extension

Výpis (1), ve kterém je uveden jeden z uzlů Extension elementu Extensions. Můžeme vidět, že nese jméno třídy obsažené ve větvi TypeName a obsahuje parametr ve větvi Using. Tato větev vyvolá událost - synchronizaci kalendáře agendy plánu návštěv na základě odběratele (Subscriber).

V dalším výpisu (2) je uzel DataWorkerUsing elementu Preprocessors. Můžeme vidět vstupní a výstupní parametry (InputParameters, OutputParameters), které jsou obsaženy v elementech

Argument. Tyto parametry jsou pak obsaženy v SQL dotazech. V Elementu TypeName je opět uvedena třída, která se zavolá při zpracování této větve.

Z definice se elementy typu Preprocessors dělí na OutputParameters a TempTables. Při běhu obou typů větví se vytvářejí dočasné entity na serveru. V případě OutputParameters se jedná o jednu atributové tabulky, TempTables zpravidla obsahují větší počet atributů a složitější podmínky v SQL dotazu.

---

```
<DataWorkerUsing>
  <DataWorker>
    <Arguments>
      <Argument>
        <Direction>In</Direction>
        <Name>InputParameters</Name>
        <Value>SubscriberId</Value>
      </Argument>
      <Argument>
        <Direction>In</Direction>
        <Name>OutputParameter</Name>
        <Value>POB_KOD</Value>
      </Argument>
      <Argument>
        <Direction>In</Direction>
        <Name>MinOutputValueCount</Name>
        <Value>1</Value>
      </Argument>
      <Argument>
        <Direction>In</Direction>
        <Name>Query</Name>
        <Value>select POBOCKA_KOD from PXD_POBOCKY
              where TG_BRANCH_ID = @SubscriberId</Value>
      </Argument>
    </Arguments>
    <TypeName>PrepareScalarDbParameterPreprocessor,
              QAS.Integration2.Data.SqlClient</TypeName>
  </DataWorker>
  <Using>Prepare Publisher</Using>
</DataWorkerUsing>
```

---

Výpis 2: Element DataWorkerUsing

### 5.1.2 Soubor Parameters

V XML elementech, které reprezentují entitu nejsou uvedeny databázové sloupce. U těchto atributů entit dochází často k modifikacím a tak je rozumné mít tyto atributy oddělené od Trandefu. Jedná se o soubor formátu XML, který obsahuje parametry uloženy formou klíč - hodnota. Při sestavení Trandefu se nahrazují tyto klíče za atributy.

V současné době má soubor 255 parametrů. Následující příklad (3) má v sobě obsažené 4 sloupce oddělené čárkou. Parametry však ve většině případů obsahují více než 20 sloupců.

---

```
<parameters>
  <parameter key="MBL_ACT_TYPES_PublisherColumns" value="A. [ACT_TYPE_CODE], A. [
    NAME_SHORT], A. [NAME], A. [VALIDITY_L]" />
</parameters>
```

---

Výpis 3: Element Parameter

### 5.1.3 Stávající řešení

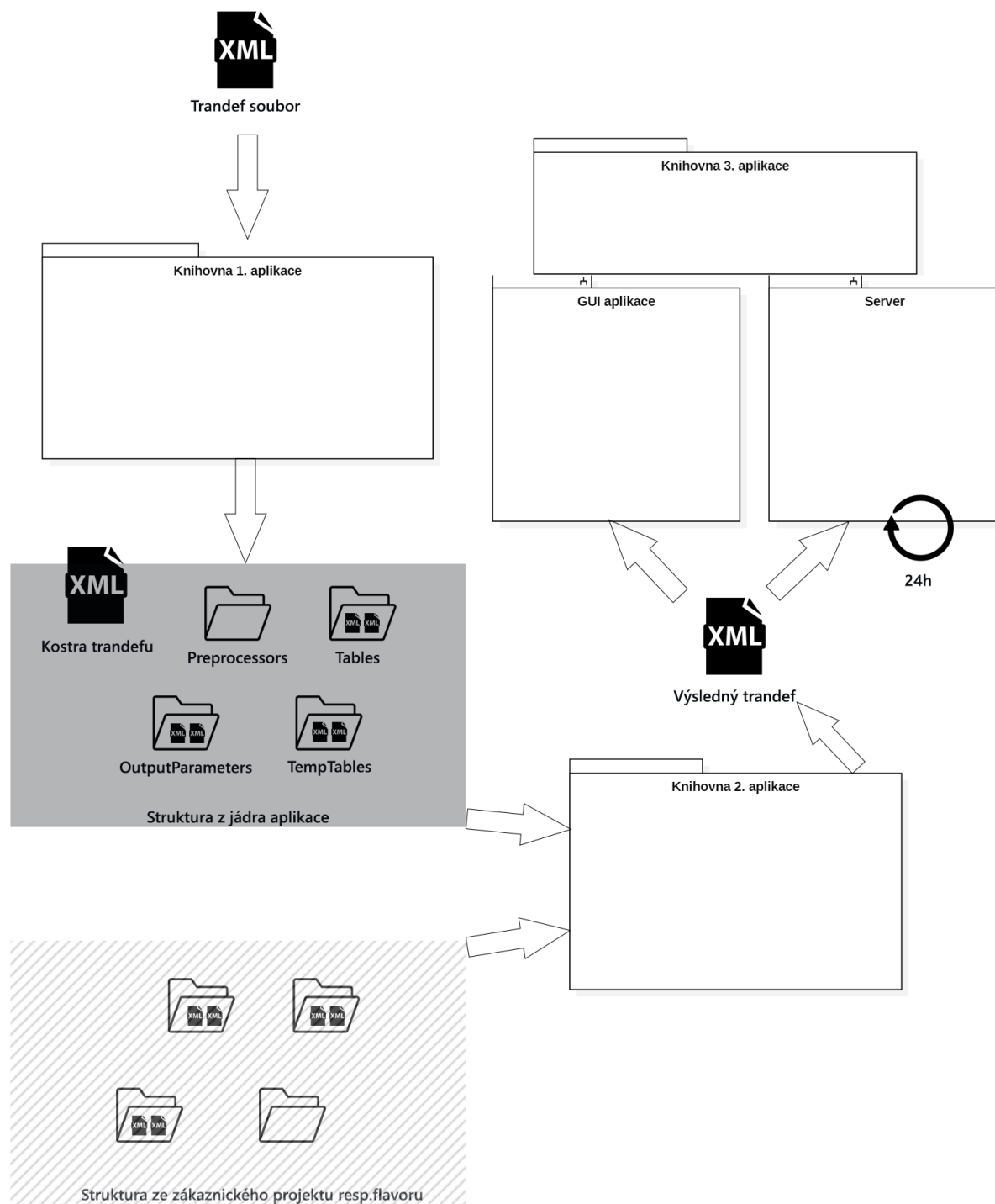
Velikost Trandefu se pohybuje mezi 7 a 8 tisíci řádky, počet se mění v závislosti na projektu. Pokud chtějí programátoři modifikovat soubor, přistupují k souboru „ručně“, to vede ke vzniku duplicit a chyb. Nelze určit, zda jde Trandef napsán správně dříve než se nasadí do samotné aplikace. Stávající proces pro práci s tímto souborem je již nadále neudržitelný.

## 5.2 Výstup

Výstup budou tvořit tři aplikace. První aplikace, která rozdělí stávající soubor do souborové a adresářové struktury. Aplikaci budu dále označovat jako Splitter. Další aplikace zajistí zpětné sloučení do souboru, kde sloučení bude ovlivněno jednotlivými projekty. Toto sloučení bude probíhat vždy při sestavení projektu. Název této aplikace bude Merger. Následné testování pomocí aplikace s GUI a testování pomocí serveru, které bude v intervalu kontrolovat funkcionality. Tato aplikace ponese název Tester.

## 5.3 Analýza a návrh

Před samotnou implementací jsem musel stanovit základní rysy a funkcionality budoucích aplikací. Celkový postup cesty z pohledu Trandefu je na obrázku 2 .



Obrázek 2: Manipulace s Trandefem

### 5.3.1 Splitter

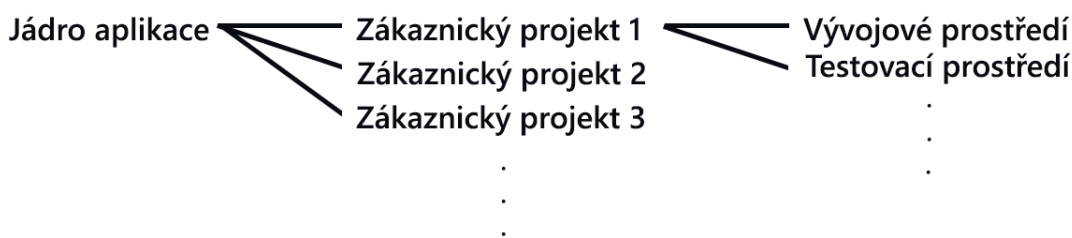
K implementaci aplikace je potřeba stanovit, jak bude vypadat adresářová a souborová struktura, do které se budou uzly ukládat. Určit, které uzly se budou ukládat do této struktury a zachovat stávající pořadí entit, protože entity na sebe navzájem navazují. Dále potřebujeme připravit soubor pro budoucí sestavení.

Pro zachování uložení pořadí entity jsem navrhl, aby každá entita nesla atribut Order, který se bude inkrementovat. Tento atribut bude sloužit pro zpětné složení souboru. Po opětovném vložení zpět do Trandefu se z větve odstraní. Aplikace pro rozdělení Trandefu se spustí pouze jednou a proto využiji standartní konzolovou aplikaci, která bude přijímat jeden parametr - cestu k umístění souboru.

### 5.3.2 Merger

Aplikace bude fungovat, jak pro sestavení Trandefu pro zákaznický projekt, který bude dědit strukturu z jádra aplikace, tak i pro samotné jádro aplikace. Každý zákaznický projekt má své prostředí, která budou také dědit ze svého projektu.

Návrh dědění by mohl vypadat následovně. Jádro aplikace, zákaznické projekty a prostředí budou obsahovat adresářovou strukturu s entitami. Tyto struktury se budou navzájem porovnávat. Struktura z prostředí bude modifikovat strukturu ze zákaznického projektu a struktura ze zákaznického projektu bude modifikovat strukturu z jádra aplikace. Pomocí tohoto porovnávání se budou entity přidávat, měnit a mazat. Aplikace bude moci sestavovat Trandef i pro samotné jádro aplikace. Bude tedy přijímat 1 až 2 parametry. Prostor si pak najde aplikace sama. Trandef se bude generovat při každém sestavení aplikace.



Obrázek 3: Struktura projektů

### 5.3.3 Tester

Aplikace musí správně převést dotazy z XML elementů do SQL dotazů. Vytvořit dočasné spojení se serverem, na kterém vytvoříme entity elementů Outprocessors a Preprocessors. Aplikace by

měla obsahovat grafické rozhraní pro snadné testování. Je žádoucí vytvořit automatický test, který bude pravidelně spouštět tento kód a tím ověřovat, zda je Trandefu v pořádku.

Aplikace bude mít funkce jako otestování celého Trandefu, jednotlivých entit, převod uzlů na SQL dotazy a testování vlastních dotazů. Její logická část bude oddělena od grafické části, tím zajistíme snadnou implementaci pro grafickou část, tak i pro část spouštěnou serverem.

## 5.4 Implementace

Již v řešení je zmíněno, že aplikace budou napsány v jazyce C# s podporou nástrojů MSBuild. Firma používá tyto prostředky pro implementaci utilit a tak mé výsledné aplikace budou zapadat do již existujícího portfolia aplikací.

### 5.4.1 Seznam tříd, stručný popis

Na obrázku 4 je třídní diagram. Pro představu uvedu stručný popis tříd.

#### **Splitter**

Konzolová aplikace pro rozdělení Trandefu do souborové a adresářové struktury.

#### **Entity, Preprocessor, OutputParameter, TempTable, Table, Argument**

Tyto třídy definují strukturu Trandefu. Nesou v sobě všechny argumenty větví. Dědicí třídy implementují funkce pro sestavení SQL dotazů.

#### **TransferDefinition**

Třída analyzuje funkce Trandefu, jehož entity si ukládá do třídních proměnných a provádí operace se jmennými prostory.

#### **AndroidDefinitionTester**

Třída, která obsahuje funkcionalitu testování Trandefu.

#### **TestAndroidDefinition**

Třída, která využívá MSBuild k testování Trandefu na serveru přičemž využívá třídu AndroidDefinitionTester.

#### **BuildMergedAndroidDefinition, BuildMergedFlavorAndroidDefinition**

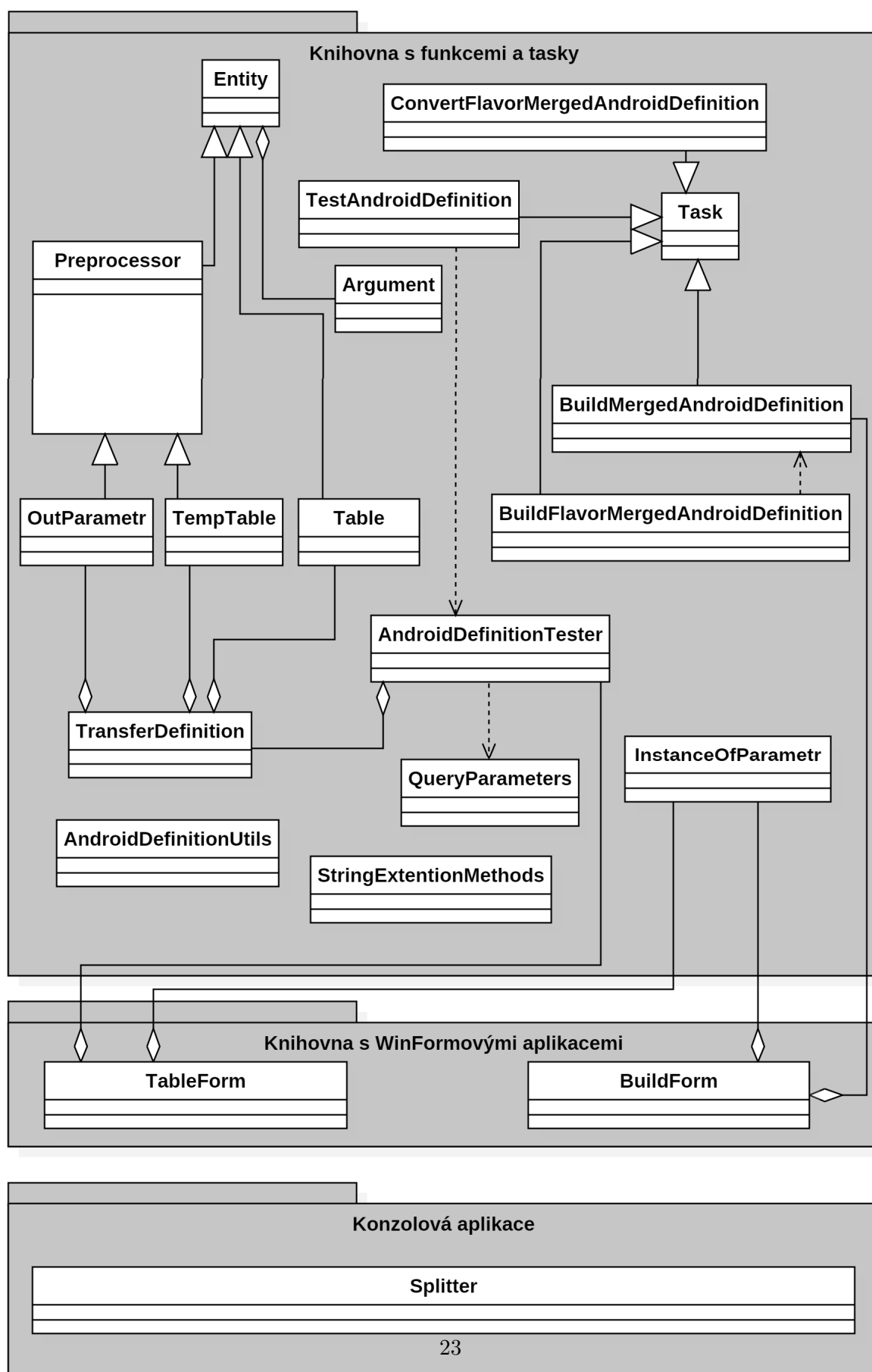
Třídy, které využívají MSBuild k složení Trandefu při sestavení projektu.

#### **InstanceOfParameter**

Třída, která si ukládá všechny entity Trandefu do TreeView, tuto třídu využívají aplikace s GUI.

#### **TableForm, BuildForm**

Třídy, které dědí z třídy Form. Jedná se o aplikace s GUI rozhraním.



### 5.4.2 Tvorba dotazů a vytváření dočasných entit na serveru

Při testování bylo nutné sestavit SQL dotazy. Každý typ entity má jinou strukturu tvorby tohoto dotazu. Např. pro elementy Preprocessors mají dotazy syntaxi `SELECT * INTO * FROM * WHERE`, tento dotaz kopíruje data do nové entity. Dotazy elementů Preprocessor se musí spustit nejdřív, tím se zajistí, že elementy typu Tables budou mít k dispozici všechny tyto dočasné entity.

Ze začátku jsem nemazal dočasné entity v domněnách, že po ukončení spojení se serverem tyto entity zaniknou. Tímto se vždy zvyšovala doba připojování na server. Tento problém jsem vyřešil mazáním dočasných entit před ukončením spojení.

### 5.4.3 Zakomponování aplikace do projektu

Pro požadavek k zakomponování sestavení Trandefu při samotném sestavení projektu jsem měl využít nástroj MSBuild. S touto technologií jsem se doposud nesetkal a tak jsem musel nastudovat, jak nástroj funguje. Využil jsem i analýzy již vytvořených projektů.

Třída, která využívá nástroje MSBuild dědí z abstraktní třídy `Microsoft.Build.Utilities.Task` a implementuje metodu `Execute()`. V této metodě se volají metody, které otestují resp. sloučí Trandef. Při sestavení projektu, který obsahuje moje aplikace, dojde k vytvoření knihovny formátu dll.

Pro spuštění tasku v zákaznické projektu jsem vložil následující kód do projektového souboru.

---

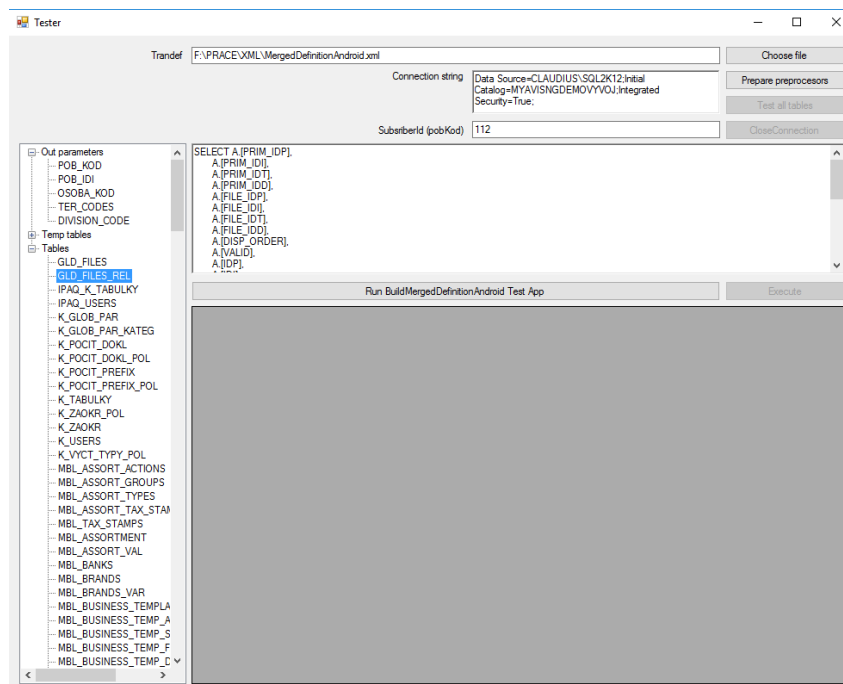
```
<Import Project="$(MSBuildExtensionsPath)\Kvados\Common\
  AndroidDefinitionTasks.targets" />
<Target Name="Build">
  <TestAndroidDefinition TrandefPath="C:\XML\MergedDefinitionAndroid.xml"
    ConnectionString="Data Source=CLAUDIUS\SQL2K12;Initial Catalog=
      MYAVISNGDEMOVVOJ;Integrated Security=True;"
    SubscriberIdOrPobKod="103"/>
</Target>
```

---

Výpis 4: Integrace MSBuildu

Ve výpisu (4) můžeme vidět import souboru targets. Soubor nese odkaz umístění knihovny dll a seznam tříd, které využívají MSBuild. Výpis dále obsahuje task, který se bude spouštět (`TestAndroidDefinition`) a jeho vstupní parametry.





Obrázek 5: GUI testovací aplikace

#### 5.4.4 Testování na serveru

Napadlo mě využít testovacího nástroje a zautomatizovat testování provedení všech dotazů. Pro testování na serveru jsem využil nástroje TFS, které toto testování umožňuje. Vytvořil jsem MSBuild task, který funguje na stejném principu jako grafická aplikace. Namísto interakce s uživatelem pomocí GUI jsem využil logování do souboru v průběhu testování. Na server bylo potřeba zkopírovat soubor s knihovnou a target soubor. A poté už jenom vše zakomponovat do TFS. Při testování se připojí na definovaný SQL server a v případě chyby v Trandefu spadne i team build.

#### 5.4.5 Grafické rozhraní

Trandef se často upravuje, proto je vhodné mít nástroj který obrovské xml vizualizuje a umožní provádět dotazy.

Grafické rozhraní testeru obsahuje vlevo seznam všech entit Trandefu uložené v TreeView. Nahoře je panel, ve kterém můžeme nastavit parametry jako je server, na který se připojíme, nebo na jaké pobočce se bude Trandef testovat. Dole je ukázaná tabulka výsledného dotazu a uprostřed je samotný dotaz, který můžeme modifikovat. Před samotným testováním je nutné spustit příkaz Prepare preprocessor, kdy se připojíme na server a vykonají se všechny dotazy Preprocessoru. Z aplikace lze také spustit další aplikaci, jejíž obsah je aplikace Merger s GUI.

## 6 Další úkoly

### 6.1 První úkol

Hned po přípravě a instalaci všech potřebných aplikací jsem pracoval na agendě „cenový průzkum“ aplikace MyAvisNG. Hlavní náplní tohoto úkolu bylo seznámení se s metodami uplatňovanými při vývoji aplikace. Vytvořil jsem WorkItem se základními prvky, které jsou často využívány v jiných agendách jako je např. filtrování či zobrazení obsahu a k němu potřebné Javovské třídy v Android Studiu.

### 6.2 Kopírování zdrojů

Zákaznické projekty využívají knihovny z jádra aplikace, tyto knihovny obsahují moduly s funkcionalitou a jsou potřebné pro spuštění zákaznických projektů.

V původním stavu bylo nutné spustit team build pro jádro aplikace a stáhnout si potřebné knihovny ze serveru. Někdy však nastane situace, že vývojáři potřebují něco upravit v jádru aplikace a vyzkoušet přímo v zákaznickém projektu. Team build trvá asi 60 minut a je lepší si skriptem knihovny do zákaznického projektu nakopírovat.

Mým úkolem bylo vytvořit utilitu, která zkopíruje knihovny z lokální verze jádra aplikace pomocí MSBuild.

Protože už jsem měl zkušenosti s nástrojem MSBuild, nebyl pro mě úkol komplikovaný. K nalezení jádra aplikace v adresářové hierarchii jsem využil rekurzivní funkci, která hledá adresář s definovaným jménem. Největší problém jsem měl s Android Studiem, kdy jsem v jádru aplikace nemohl vygenerovat soubory aar. Pomocí gradle tasků. Problém jsem však vyřešil a utilitu integroval do bat souboru, který využívá MSBuild.

### 6.3 Rozvoj bezpečnosti

Jako tým jsme měli za úkol udělat množinu kroků, které povedou ke zlepšení bezpečnosti mobilní aplikace. Jednalo se o zašifrování databáze, detekce rootu zařízení a přesun uložště včetně SQLite databáze z externího uložště do interního. A právě poslední z těchto úkolů jsem měl na starost.

Součástí implementace byla změna služeb WorkItemu aplikace, ze stávající služby, která využívala externí uložště na nové rozhraní. A přidání rozhraní k přesunutí souborů. V zadání bylo určeno, že vývojářské prostředí nebude přesouvat databázi. A tak jsem vytvořil třídu, která rozšiřuje funkcionalitu rozhraní pro přesun dat. Tuto třídu jsem implementoval do všech prostředí a ve vývojářské větvi jsem naimplementoval vynechání přesunu databáze. Dále jsem

do manifestu aplikace implementoval Content providera pro umožnění otevírání souborů naší aplikace v jiných aplikacích. Součástí bylo také definování cest, které Content provider bude moct poskytnout jiným aplikacím. Poslední změnou byla změna funkcionalit Javovských tříd, které pracovaly s ukládáním do uložště. Této části jsem věnoval nejvíce času.

Ze zmiňovaných bezpečnostních kroků se v současné době používá tento přesun a ukládání dat do interního uložště. Práce se zašifrovanou databází je příliš pomalá. Detekce rootu není nikdy 100% rozeznatelná.

## 6.4 Budoucí vývoj aplikací

V současné době manažeři a obchodní zástupci používají převážně produkty firmy Apple. Ať už z pocitu luxusu, dlouholeté věrnosti značce nebo jen pro to, že si to můžou dovolit. Zároveň do zařízení s iOS se lze hůř dostat a prolomit jeho bezpečnostní prvky.

Tím, že firma v současné době vyvíjí mobilní aplikace pouze pro platformu Android, je možné že se staví do nevýhody oproti konkurenci. Navíc metody vývoje formou WinForms SmartClient aplikací byly pro mě velmi těžce naučitelné a nejsou to nejnovější technologie na poli vývoje rozsáhlých aplikací.

Jednou z možných alternativ by byl vývoj prostřednictvím technologie Xamarin, která umožňuje vyvíjet aplikace multiplatformně s podporou jazyka C#. MyAvisNG by tak mohly využívat zařízení s Android, iOS i Windows. Xamarin obsahuje službu TestCloud, která umožňuje automatické testování aplikace. Jeho součástí je také platforma, která analyzuje nasazení aplikace, což umožní sledovat veškeré problémy na straně zákazníků.

## 7 Využité dovednosti získané v průběhu studia

Ačkoliv syntaxe různých programovacích jazyků je vždy odlišná, principy programování se liší jen zřídka. A právě programování ať už v jakémkoliv jazyce tvoří velmi pevný základ při studiu nových technologií, které jsem potřeboval při odborné praxi. Zkušenosti v programování jsem získal především ze všech předmětů týkajících se programování, kde jsem docházel na praktické cvičení či jsem vypracoval semestrální projekt.

Při psaní kódu jsem využil znalosti z předmětů *Programování I* a *II* k získání všeobecných základů principů a paradigmat programování, předmětu *Programovací jazyky I*, kde jsem využíval syntaxi jazyku pro psaní Java tříd v Android Studiu, předmětů *Programovací jazyky II* pro psaní utilit ve Visual Studiu a *Algoritmy I* a *II*, které prohlubují znalosti všeobecného programování.

Předměty *Úvod do databázových systémů* a *Databázové a informační systémy* jsem užil při sestavování SQL dotazů a k orientaci v databázovém systému aplikace. Další z předmětů, který mi velmi pomohl, je *Tvorba aplikací pro mobilní zařízení II*, tento předmět mi poskytl znalosti a zkušenosti k vývoji aplikacím na platformě Android.

Při vývoji se používaly metody, které byly probírány v předmětu *Úvod do softwarového inženýrství* a *Vývoj informačních systémů*.

## 8 Chybějící znalosti

V průběhu praxe jsem pracoval s TFS, což se řadí mezi práci s integrovanými systémy pro správu kódu. S těmito systémy jsem se na škole nesetkal a tak jsem čerpal znalosti z postupů mých kolegů a oficiální dokumentace. Další z věcí, kterou jsem neznal je MSBuild, která je součástí .NET frameworku. V průběhu studia jsem neměl žádný volitelný předmět, který by se na pokročilejší funkce v .NET zaměřil.

Jsem si jist, že nelze pokrýt všechny aspekty informatiky v průběhu 3-letého studia. Nicméně škola mě naučila základní principy programování, ze kterých jsem stavěl při učení nových jazyků a technologií.

## 9 Závěr

Jsem rád, že jsem se mohl podívat do firmy, kde se vyvíjí software a sledovat postupy a metody při jeho tvorbě. V průběhu praxe jsem udělal řadu řešení, zejména celkovou optimalizaci s Trandefem, což byl můj hlavní cíl při odborné praxi. Tyto řešení bude firma využívat. Zjistil jsem, že se velmi liší práce programátora ve společnosti a studenta, který doma nebo ve škole programuje pro získání znalostí nebo jen, protože ho to baví. Při vytváření produktu pro zákazníka má veškerý vývoj jasné hranice a je minimalizována kreativita programátora. Tento fakt mě ovlivní při volbě budoucího zaměstnání.

Absolvování odborné praxe ve firmě Kvados a.s. hodnotím kladně. Zejména jsem se vylepšil v programování. Poznal jsem principy vývoje softwaru v praxi, jako je metodika agilního vývoje Scrum nebo problematiku práci v týmu.

## Literatura

- [1] Kvados a.s. [online] [cit. 2017-03-29], Dostupné z: <http://kvados.cz/>
- [2] W3Schools Online Web Tutorials [online] [cit. 2017-03-29], Dostupné z: <https://www.w3schools.com/>.
- [3] Výuka pro vývojáře na webu Microsoft Developer Network | MSDN [online] [cit. 2017-03-29], Dostupné z: <https://msdn.microsoft.com/>.